Depth Measurement and 3D Reconstruction of the Scene from Multiple Image Sequence

Thesis submitted to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology Dual Degree

by

V Tulasi Krishna

(Roll- 710CS1157)

under the supervision of

Prof. Ramesh Kumar Mohapatra

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela- 769008, India

May, 2015

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Prof. Ramesh Kumar Mohapatra
Assistant Professor

June 1, 2015

Certificate

This is to certify that the thesis entitled **Depth Measurement and 3D Reconstruction of the Scene from Multiple Image Sequence** by **V Tulasi Krishna** is a record of work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of **M.Tech. Dual Degree in Computer Science and Engineering**. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Ramesh Kumar Mohapatra

Acknowledgements

I would like to express my special appreciation and thanks to my supervisor Professor R. K. Mohapatra who has been a great mentor for me. I thank him for his valuable suggestions.

I am grateful to Prof. Sunil Kumar Sarangi, Director, National Institute of Technology, Rourkela for providing excellent facilities in the Institute for carrying out research.

I am thankful to the Head of the Department and professors of department for their valuable help during the preparation of this work.

There are no words to explain how grateful I am to my parents and my sister for all of their sacrifices and prayers that they made for me. I also take this opportunity to thank all of my friends who helped me during the preparation of this work.

V Tulasi Krishna

Roll No. : 710CS1157

June 1, 2015

Abstract

In the everyday life we continuously see many scenes and easily recognize them and perceive their 3D structure, this simple 3D scene reconstruction problem is a challenging task in the area of computer vision from the past few decades. In the present work, we explored few options to tackle this problem and try to establish the bound of these methods. In this project we, input the image sequence of the stationary scene from different views and get the depth of the image from the camera from which we ultimately reconstruct the 3D scene. We use epipolar geometry to find the fundamental matrix and match correspondences between images, we calibrate the camera intrinsic parameters and use them triangulate to get the depth of the object. An efficient dense matching has been implemented to increase the number of correspondences between a pair of image. This algorithm takes few correspondences the matches with highest some points of interest which have the highest gradient as the initial seed points and propagates the matches in the neighborhood of seed points from the most gradient pixels to less gradient ones. We also analyzed the increase in the number of the points by applying dense matching algorithm.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 3D scene reconstruction

3D reconstruction is one of the areas in computer vision which has scope for significant improvement and the existing solutions are not completely satisfactory. With the swift progress in computer technology, people have huge demand for a 3D modeling. It has huge demand in the areas of

1. Movies
2. Games
3. Virtual reality
4. Space research
5. Real time battle support
6. Robotics

Many other usage scenarios may present themselves in near future and a proper 3D model should be constructed as soon as possible.

The 3D model is the reconstruction of the 3D scene from the images taken of the given scene. It has the information about the depth, height and width of the scene. We can see this 3D scene in many views.

In the present day we use 3D scanner for this purpose which uses Lasers to scan the depth of the object from the camera. But this process is highly costly and cannot be implemented on all the object i.e., if we have a huge building we cannot scan it with the 3D scanner. To avoid this problem we started trying to get the 3D modelling from the images taken from the camera which are easily available, more precise and low expense.

3D reconstruction form image sequence is based on the key concept of reconstructing 3D point cloud from two images, we use images taken from the camera for getting 3D reconstruction of the images. For this the intrinsic parameters of the camera are needed to be calculated. Feature extraction and feature matching plays a crucial role in the reconstruction problem. Good features give good correspondences which are mapped in the 3D cloud, so feature extraction is an important step The most frequently used feature detection algorithms are Harris and SIFT feature detection. Harris corners are more sensitive to the noise which produces outliers and is affected by the scale of the images. The SIFT features are Invariant to Scale and partially invariant to the rotation, so SIFT features is used to matching correspondences in image pairs. In succession, we use epipolar geometry which gives the fundamental matrix that relates the right-image and left-left image. The fundamental matrix used to relate the points in two images of the same scene, it is calculated using the epipolar geometry. The essential matrix is used to find the camera coordinates in the world plane with respect to the given scene, it has the information about the rotation and the translation of the two cameras with respect to each other. The essential matrix is calculated from the fundamental matrix and the intrinsic parameters of the camera. With the correspondences and the essential matrix we can triangulate image points to the 3D points. These corresponding points are very few to build the 3D model so we will increase the number of 3D points by using dense match algorithm proposed by[8]. After which we triangulate the dense points to get the dense cloud.

1.2 Objective

1. The main objective of this project is to analyze the problem regarding the number of the matched correspondences to compute the 3D points.

2. Implementing the dense matching algorithm on pair of images to increase the number of matches

3. To reconstruct 3D point cloud from the matched correspondences after dense matching

4. To compare the dense matches to the initial matches and analyze the significance of the dense matching.

CHAPTER 2

OVERVIEW

2.1 Introduction

The following figure.2.1 shows the sequence of steps involved in constructing a 3D scene from a set of image sequences which are fed to the algorithm at the start.

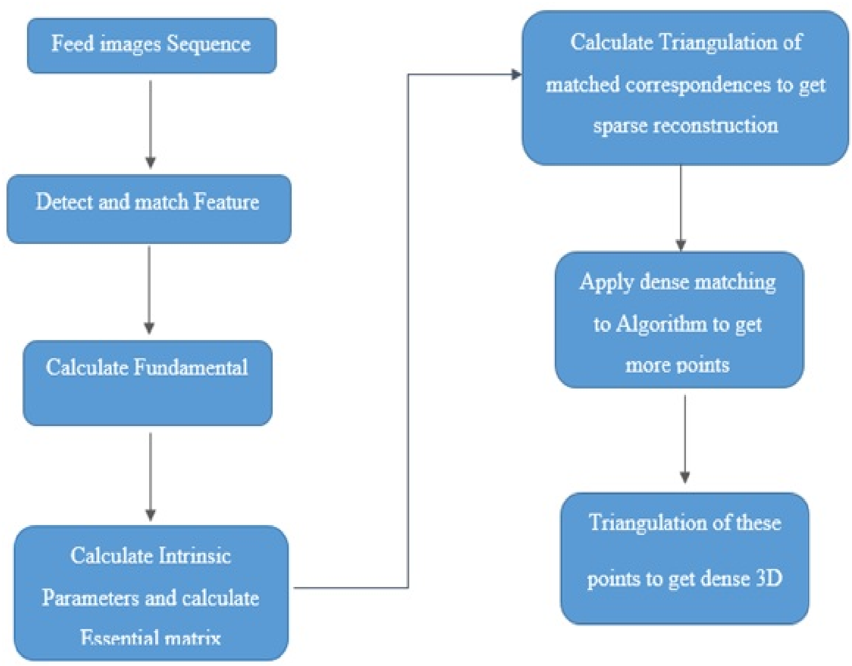


Figure 2.1: Overview of the Reconstruction Process

In this thesis we mainly see about the application of dense matching algorithm and its results on the output we get.

Chapter 3

METHODOLOGY

3.1 Extraction and matching of feature points:

Extraction and matching of feature points is one of the important steps in reconstructing 3D scenes as precision of this step directly results in the accuracy of the 3D points. Different Feature points are generated and matched using the SIFT algorithm. D.G.Lowe in 1999 proposed the SIFT algorithm [9]. SIFT features are invariant to scaling, transformations or rotations in the image domain and robust. Steps of the SIFT algorithm: Extreme value point detection in the space of scale; Extraction of accurate positioning extreme value point; Specification of direction parameter for each key point; Generation of key point descriptive clauses[4].

The basic idea of the SIFT features is to construct different scales of image by applying Gaussian function and repeatedly smoothing and subsampling the input image, then these scales are substacted from one another to get the local extrema which are taken as the SIFT feature points. These interest points are then stored as the descriptors. These SIFT descriptor are matched using best bin first approximation is used to get the correspondence points[11].

3.2 Calculating fundamental matrix

The 3D point **X** projected onto an image plane as a 2D point **x** is represented by 3.2.1

$$x = P \times X \tag{3.2.1}$$

Where

$\mathbf{X} = (\mathbf{X},\ \mathbf{Y},\ \mathbf{Z},\ \mathbf{1})^T$ is a 3D point represented as an homogeneous 4-vector,

$\mathbf{x} = (\mathbf{x},\ \mathbf{y},\ \mathbf{1})^T$ is the projection of **X** in image plane represented as an homogeneous vector,

and P is a projection matrix of dimension 3×4, as shown below in figure.3.1.

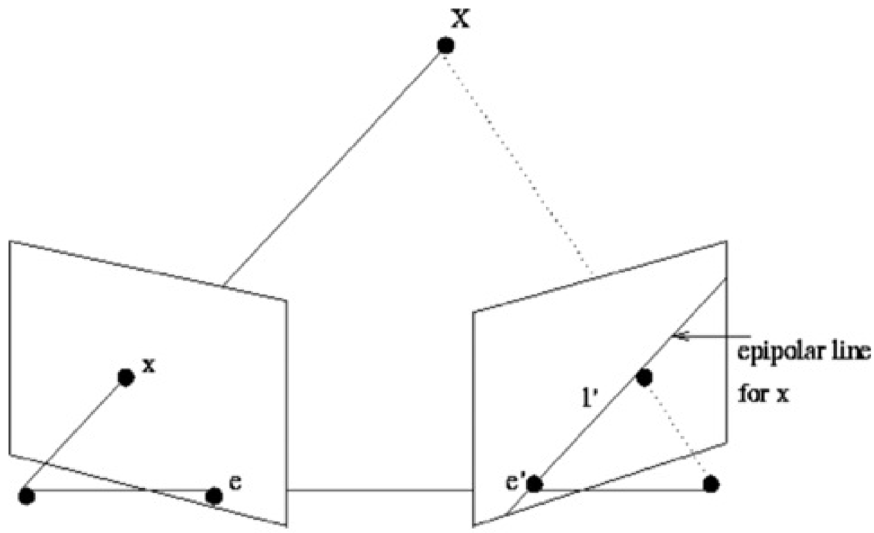


Figure 3.1: Explaining Epipolar Geometry

The point $x^{'}$ from the right image corresponding to the point x in the left image should lie on a line called epipolar line. The epipolar geometry concerns with the geometric constraints between to two views of the single scene. Here, the epipoles are e, $e^{'}$ corresponding to two images. The epipoles do not depend on the structure of the scene rather they depend on the relative pose of the camera. The fundamental matrix algebraically represents the epipolar geometry. A fundamental matrix F is a unique homogenous matrix of dimension 3×3 and of rank 2 satisfying

$$x'^{T} \times F \times x = 0 \tag{3.2.2}$$

Only the corresponding points from left and right images can be used to calculate the fundamental matrix for every corresponding point $x_i \leftrightarrow x_i^{'}$

3.2.1 8-point algorithm for constructing fundamental Matrix

The equation 3.2.2 can also be written as

$$[xx', yx', x', xy', yy', y', x, y, 1]f = 0 \tag{3.2.3}$$

where

$$\boldsymbol{x} = (x, y, 1)^{T} and, \boldsymbol{x'} = (x', y', 1)^{T} \tag{3.2.4}$$

The equation of 8 points is given by

$$\begin{bmatrix}
{x^1}_1{x^1}_2 & {x^1}_1{y^1}_2 & {x^1}_1 & {y^1}_1{x^1}_2 & {y^1}_1{y^1}_2 & {y^1}_1 & {x^1}_2 & {y^1}_2 & 1 \\
{x^2}_1{x_2}^2 & {x^2}_1{y^2}_2 & {x^2}_1 & {y^2}_1{x_2}^2 & {y^2}_1{y^2}_2 & {y^2}_1 & {x^2}_2 & {y^2}_2 & 1 \\
{x^3}_1{x_2}^3 & {x^3}_1{y^3}_2 & {x^3}_1 & {y^3}_1{x^3}_2 & {y^3}_1{y^3}_2 & {y^3}_1 & {x^3}_2 & {y^3}_2 & 1 \\
{x^4}_1{x_2}^4 & {x^4}_1{y^4}_2 & {x^4}_1 & {y^4}_1{x^4}_2 & {y^4}_1{y^4}_2 & {y^4}_1 & {x^4}_2 & {y^4}_2 & 1 \\
{x^5}_1{x_2}^5 & {x^5}_1{y^5}_2 & {x^5}_1 & {y^5}_1{x^5}_2 & {y^5}_1{y^5}_2 & {y^5}_1 & {x^5}_2 & {y^5}_2 & 1 \\
{x^6}_1{x^6}_2 & {x^6}_1{y^6}_2 & {x^6}_1 & {y^6}_1{x^6}_2 & {y^6}_1{y^6}_2 & {y^6}_1 & {x^6}_2 & {y^6}_2 & 1 \\
{x^7}_1{x^7}_2 & {x^7}_1{y^7}_2 & {x^7}_1 & {y^7}_1{x^7}_2 & {y^7}_1{y^7}_2 & {y^7}_1 & {x^7}_2 & {y^7}_2 & 1 \\
{x^8}_1{x^8}_2 & {x^8}_1{y^8}_2 & {x^8}_1 & {y^8}_1{x^8}_2 & {y^8}_1{y^8}_2 & {y^8}_1 & {x^8}_2 & {y^8}_2 & 1
\end{bmatrix}
\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

$$A_8 \times f = 0 \tag{3.2.5}$$

Where A_8 is matrix of size 8 x 9. We get some outliers in the correspondences due to noise. We get F by least square method 3.2.6

$$\min_f (\| A_8 f \|^2) \tag{3.2.6}$$

We compute fundamental matrix **F** by solving a set of linear equations using least squares method. As we contain outliers in correspondences 8-point algorithm we do not get an accurate solution for fundamental matrix. To remove this problem we use more points of correspondence and apply RANSAC, a line fitting algorithm. Fishler and Bolles proposed Random Sampling Consensus (RANSAC) algorithm[1].

RANSAC follows the steps mentioned below.

Let there be n correspondences, which we got from feature matching.

1. For all combinations of eight correspondences from n correspondences
   (a) Compute a solution for F using these 8 point algorithm.
   (b) Count number of inliers. Inliers are the correspondences which are not more 2 pixel distance away from its epipolar line which we get by substituting x and x.
2. Choose the **F** with the largest inliers.

By this way we can get a reliable fundamental matrix **F** which also eliminate the outliers from the matched features by SIFT matching.

3.3 Camera calibration

By taking a planar object with known dimensions an taking pictures of this object in different views with respect to the camera we can get intrinsic parameters of the camera [17],[6]. We use the checker board pattern as the planar image with the size of the square know we can calculate the camera matrix. The checkerboard used for our purpose is given below in the figure.3.2.

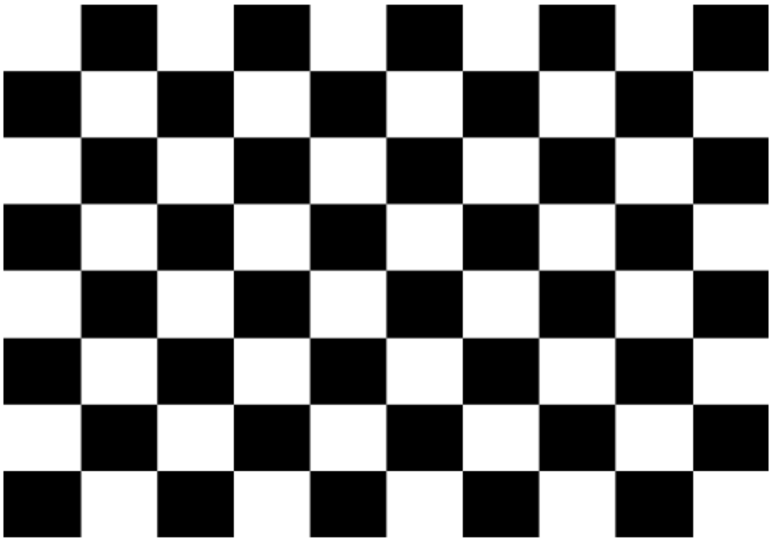


Figure 3.2: Image for Camera Calibration

Procedure for camera calibration:

1. By using the harris feature corner detection algorithm we find the corners in the images.
2. The linear line are estimated such that they pass through the corner.
3. The corners are computed as the points where two line crosses each other.
4. We take the bottom left corner as the checker board pattern as the origin of the world coordinates and use the distance between two corners as the

constraint to calculate the camera matrix

By this way we can get the intrinsic parameters of the camera **K** .This value of **K** is used in finding the essential matrix.

3.4 Essential matrix

After getting the intrinsic parameters of the camera matrix we can directly get the essential matrix from the equation 3.4.1.

$$E = K' \times F \times K \tag{3.4.1}$$

The above equation is taken from reference [4]. By taking first camera as the origin of the world coordinate system and decomposing the essential matrix E we will get rotation and translation matrices of second camera with respect to the first camera.

From these above values we can triangulate the correspondences from matched features to get sparse 3D point cloud.

3.5 Dense matching

The number of points we get in the sparse reconstruction are very less because the when we check for the correspondences in two images we only take the textured points and ignore the points surrounding this local maxima. Due to which a lot of the points are missed to avoid this we will use dense matching algorithm.

The main idea of the algorithm is to initiate matching few points of interest which have the highest gradient as seed points and then spreading the matches in the neighborhood of seed points having highest gradient pixels to less gradient ones. This algorithm is divided into two steps: seed selection and propagation.

3.5.1 Selection

We use ZNCC (zero-mean normalized cross-correlation) correlation measure to match seeds as it is invariant to linear radiometric changes.

$$ZNCC(x_1, x_2) = \frac{\sum_i (I(x_1+i) - \bar{I}(x_1))((I(x_2+i) - \bar{I}(x_2))}{\sqrt{\sum_i (I(x_1+i) - \bar{I}(x_1)^2 \sum_i (I(x_2+i) - \bar{I}(x_2)^2}} \tag{3.5.1}$$

The score 3.5.1 gives the relationship between two image patches **x1** and **x2**. The score of ZNCC ranges from -1 to 1. A value of 1 indicates a perfect match. If the score is greater than the threshold than the points are considered to be matches if not mismatches.

3.5.2 Propagation

The second step of the dense matching is to increase the number of matches seeds and by propagating the initial seed matches. The seed with best ZNCC score 3.5.1 is deleted from the queue and it neghbourhood is searched for possible matches. If the neighbours satisfy the given constrain then they are added to the seed matches and this procedure is repeated until the seeds become empty. By this way we can spread the area of the matches around the seeded points and significantly increase the number of matches between two images. The algorithm[8] concerning this procedure is written below.

Algorithm 1 Dense Matching Algorithm

Input: Seed

Output: Densemap

```
Seed the initial matches from SIFT matching
while Seed != Null do
    pop the best match (a, b) from Seed
    temp != Null
    //(Push the new candidate matches in temp)
    for (u, v) in Neighbour(x, y) do
        if (u,*) and (*,v) not in Densemap and f(u)>t, f(v)>t and ZNCC (u,v)>.8) then
            push match (u, v) in temp
        end if
    end for
    //(Updating Seed and pushing good candidate matches to Densemap)
    while temp != Null do
        pop the best match (u, v) from temp
        if (u,*) and (*,v) not in Densemap then
            push match (u, v) in Densemap and Seed
        end if
    end while
end while
```

Where f(a)=maximum $\{I$(a+i)-I(a)—,i ϵ $\{$(0,1),(1,0),(0,-1),(-1,0)$\}\}$, f(a) gives the measure of the surface gradient for the pixel at a, this value is used to stop propagation of the area, which is not matchable. If f(a) $<$ t where t $=$ 0.01 then the propagation is stopped. Here Seed is the input, Densemap is the output and Neighbour(x,y) is set of the neighbours of x and y in corresponding images

Chapter 4

Implementation

4.1 Inputs

4.1.1 Input images for 3D reconstruction

The figures.4.1,4.2,4.3 below are taken from the mobile camera of 8 megapixel resolution whose camera is calculated by us.

Figure 4.1: Input images of first view and second view

Input Image - 3

Input Image - 4

Figure 4.2: Input images of third view and fourth view

Input Image - 5

Figure 4.3: Input image of the fifth view

4.1.2 Input images for camera calibration

The figure 4.4 below shows the images used to calculate the camera calibration of the mobile camera that is used for 3D reconstruction.

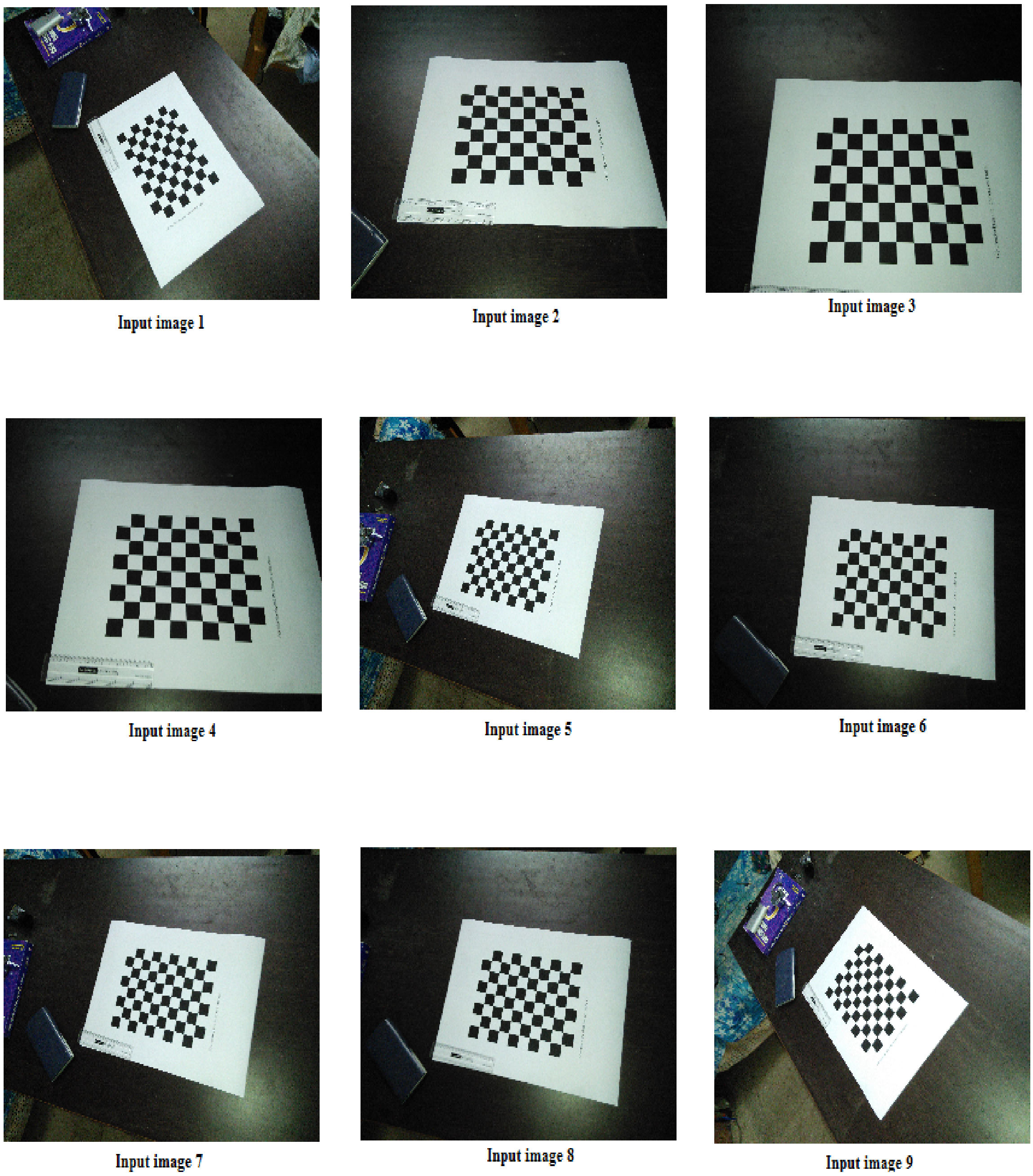
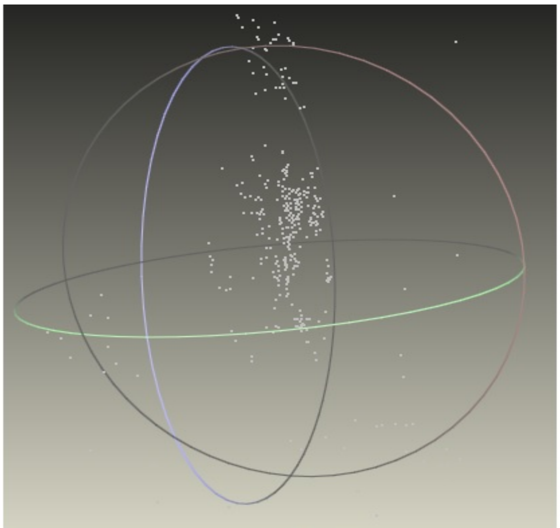


Figure 4.4: Some of the input images used to calculate intrinsic camera parameters

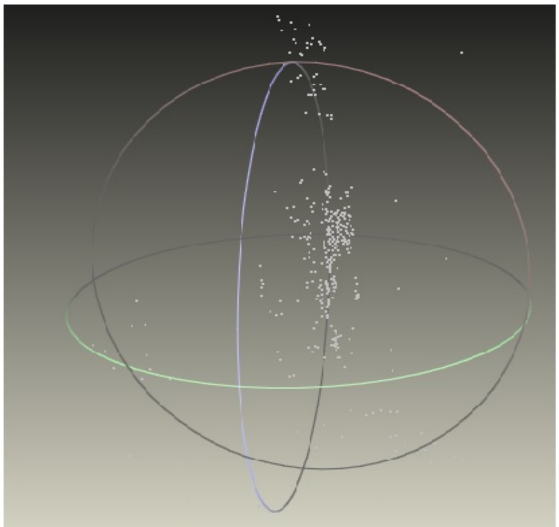
4.2 Output

4.2.1 Sparse reconstruction

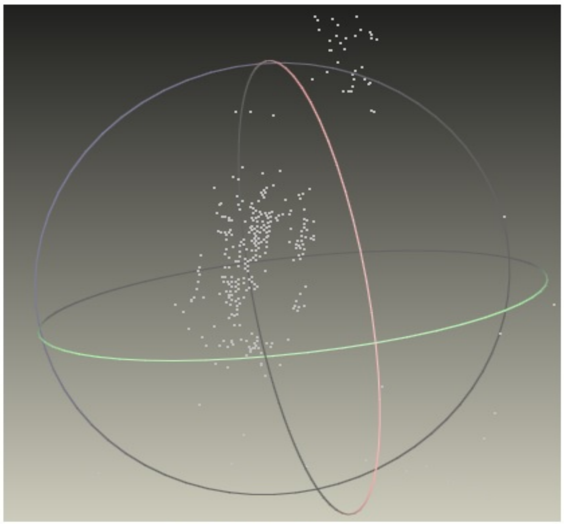
We get the sparse point cloud, Figure.4.5, from the correspondences which we get from feature matching using SIFT before applying dense algorithm



Sparse point cloud - 1



Sparse point cloud - 2



Sparse point cloud - 3

Figure 4.5: Sparse point cloud viewed in three different angles

4.2.2 Applying dense matching

The below figures.4.6,4.7,4.8,4.9 describe the dense map explained in the dense matching algorithm.

Figure 4.6: Dense matching for the Image pair 1 and 2

Figure 4.7: Dense matching for the Image pair 2 and 3

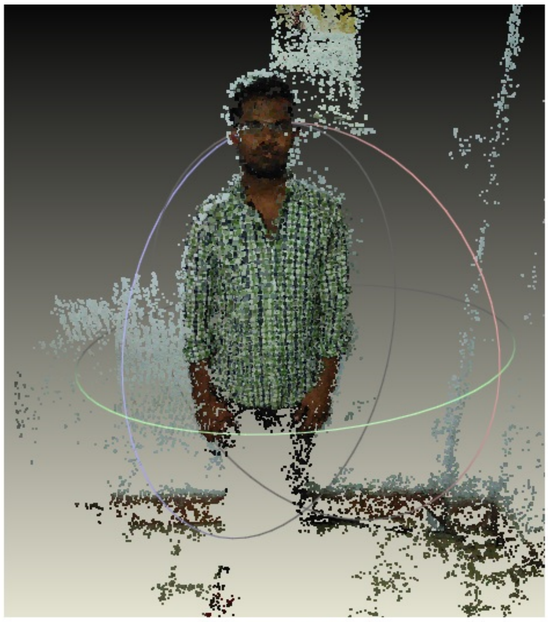
Figure 4.8: Dense matching for the Image pair 3 and 4

Figure 4.9: Dense matching for the Image pair 4 and 5

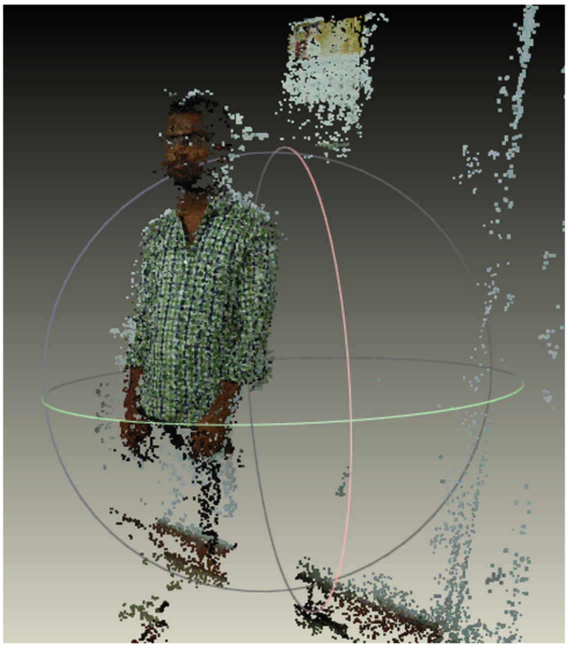
4.2.3 After applying dense matching

The figures.4.10 describes the 3D point cloud in different views of the singe scene.

Dense point cloud - 1



Dense point cloud - 2



Dense point cloud - 3

Figure 4.10: Dense point cloud viewed in 3 different angles

4.2.4 Comparing both number of points

In the below figure.4.11,4.12 the red dots are the 3D points are the initial matches before the dense matching and the blue dots represents the 3D points we got from the dense matching which are significantly more than initial matches.

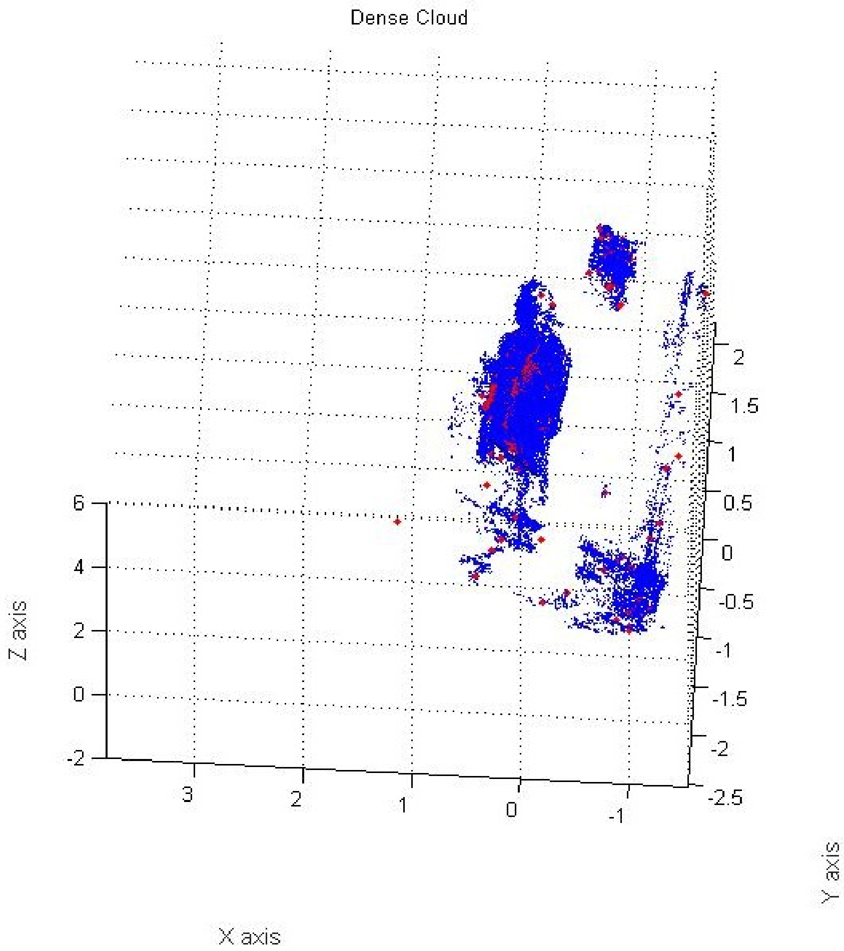


Figure 4.11: 3D point cloud generated in matlab view 1

This is the same dense point cloud which is viewed in different angle in the matlab.

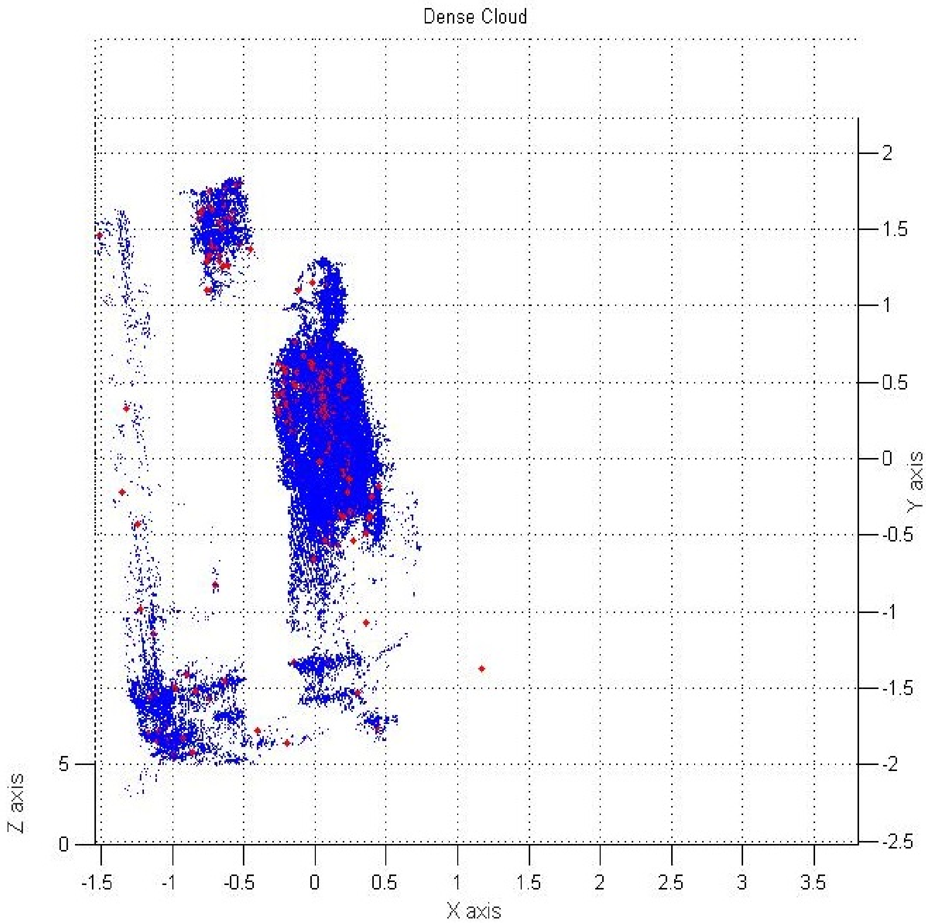


Figure 4.12: 3D point cloud generated in matlab view 2

4.2.5 Intrinsic parameters of the camera

The parameters of the camera matrix K is calculated by taking the checkerboard images as the input mentioned in 3.3 gives the following values of the output.

$$K = \begin{pmatrix} 3503.92335468781 & 0 & 0 \\ 0 & 3276.25002894256 & 0 \\ 1557.06940894345 & 934.240378256405 & 1 \end{pmatrix}$$

4.2.6 Table containing no of sparse and dense matches

This table.4.1 represents the number of the sparse matches and dense matches for different set of input images taken from mobile camera whose intrinsic parameters are calculated.

Input no	No of sparse matches	No of Dense matches
1	352	128816
2	162	97307
3	153	129187
4	148	20628
5	116	85761
6	89	43716
7	78	69421
8	24	3235

Table 4.1: Table containing the number of sparse and dense matches of given input set

The plot in the below figure.4.13 shows that there is no dependence of the dense points on the number of the sparse points

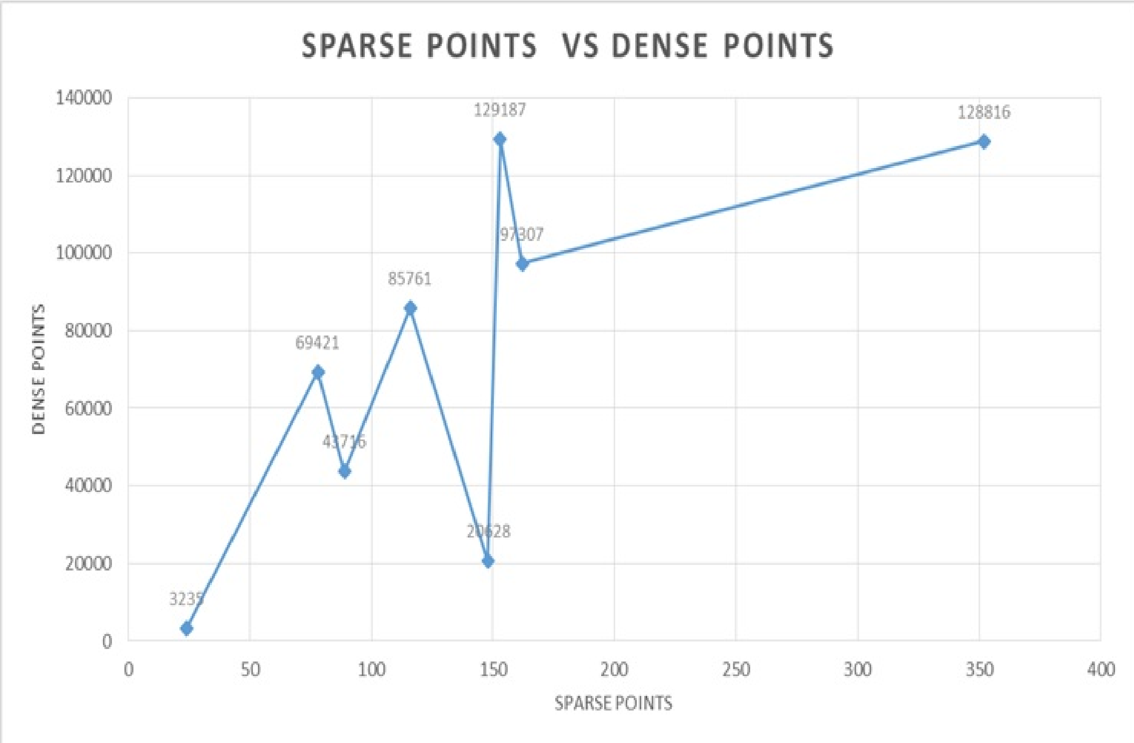


Figure 4.13: Graph describing sparse points and dense points

CHAPTER 5

Conclusion and Future work

Thus, a 3D scene reconstruction has been done successfully from the mentioned image sequences. The Dense Matching Algorithm has been implemented in MATLAB. Applying this algorithm significantly improves the number of 3D points in the output. Though the dense algorithm takes the sparse points as the seeds the number of the final points we get doesnt depend on the feature of the image. For some inputs, only few good seed match is enough to start an avalanche for matches in the entire region of the image.

Though this algorithm gives significant number of 3D points takes a lot of time to compute. This computational time can be reduced using improvised algorithms. The texture mapping can be done to the given 3D point cloud.

Bibliography

[1] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[2] Furukawa, Y. and Ponce, J., "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 32, no. 8, pp. 1362–1376, 2010.

[3] Goesele, M., Curless, B., and Seitz, S. M., "Multi-view stereo revisited," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2402–2409, IEEE, 2006.

[4] Hartley, R. and Zisserman, A., "Multiple view geometry in computer vision (cambridge university, 2003)," *C1 C3*.

[5] Hartley, R. I., "Theory and practice of projective rectification," *International Journal of Computer Vision*, vol. 35, no. 2, pp. 115–127, 1999.

[6] Heikkila, J. and Silvén, O., "A four-step camera calibration procedure with implicit image correction," in *IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition, 1997. Proceedings, pp. 1106–1112, IEEE, 1997.

[7] Kuffar, H. and Takaya, K., "Depth measurement and 3d metric reconstruction from two uncalibrated stereo images," in *CCECE 2007. Canadian Conference on Electrical and Computer Engineering*, pp. 1460–1463, IEEE, 2007.

[8] Lhuillier, M. and Quan, L., "Robust dense matching using local and global geometric constraints," in *Proceedings of 15th International Conference on Pattern Recognition*, vol. 1, pp. 968–972, IEEE, 2000.

[9] Lowe, D. G., "Object recognition from local scale-invariant features," in *The proceedings of the seventh IEEE international conference on Computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.

[10] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] Muja, M. and Lowe, D. G., "Fast approximate nearest neighbors with automatic algorithm configuration.," *VISAPP (1)*, vol. 2, 2009.

[12] Peng, K., Chen, X., Zhou, D., and Liu, Y., "3d reconstruction based on sift and harris feature points," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 960–964, IEEE, 2009.

[13] Taillandier, F. and Deriche, R., "Reconstruction of 3d linear primitives from multiple views for urban areas modelisation," *INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY REMOTE SENSING AND SPATIAL INFORMATION SCIENCES*, vol. 34, no. 3/B, pp. 267–272, 2002.

[14] Ze-tao, J., Bi-na, Z., Min, W., and Wen-huan, W., "A fully automatic 3d reconstruction method based on images," in *WRI World Congress on Computer Science and Information Engineering*, vol. 5, pp. 327–331, IEEE, 2009.

[15] Ze-tao, J., Bi-na, Z., Min, W., and Zhong-xiang, C., "A 3d reconstruction method based on images dense stereo matching," in *WGEC'09. 3rd International Conference on Genetic and Evolutionary Computing*, pp. 319–323, IEEE, 2009.

[16] Zhang, Y.-K. and Xiao, Y., "A practical method of 3d reconstruction based on uncalibrated image sequence," in *9th International Conference on Signal Processing, 2008. ICSP 2008.*, pp. 1368–1371, IEEE, 2008.

[17] Zhang, Z., "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 22, no. 11, pp. 1330–1334, 2000.